# MWU-aware Part-of-Speech Tagging with a CRF model and lexical resources

Matthieu Constant, Anthony Sigogne

# MWU-aware Part-of-Speech Tagging with a CRF model and lexical resources

**Matthieu Constant**
Université Paris-Est, LIGM
5, bd Descartes - Champs/Marne
77454 Marne-la-Vallée cedex 2, France
mconstan@univ-mlv.fr

**Anthony Sigogne**
Université Paris-Est, LIGM
5, bd Descartes - Champs/Marne
77454 Marne-la-Vallée cedex 2, France
sigogne@univ-mlv.fr

## Abstract

This paper describes a new part-of-speech tagger including multiword unit (MWU) identification. It is based on a Conditional Random Field model integrating language-independent features, as well as features computed from external lexical resources. It was implemented in a finite-state framework composed of a preliminary finite-state lexical analysis and a CRF decoding using weighted finite-state transducer composition. We showed that our tagger reaches state-of-the-art results for French in the standard evaluation conditions (i.e. each multiword unit is already merged in a single token). The evaluation of the tagger integrating MWU recognition clearly shows the interest of incorporating features based on MWU resources.

## 1 Introduction

Part-of-speech (POS) tagging reaches excellent results thanks to powerful discriminative multi-feature models such as Conditional Random Fields (Lafferty et al., 2001), Support Vector Machine (Giménez and Márquez, 2004), Maximum Entropy (Ratnaparkhi, 1996). Some studies like (Denis and Sagot, 2009) have shown that featuring these models by means of external morphosyntactic resources still improves accuracy. Nevertheless, current taggers rarely take multiword units such as compound words into account, whereas they form very frequent lexical units with strong syntactic and semantic particularities (Sag et al., 2001; Copestake et al., 2002) and their identification is crucial for applications requiring semantic processing. Indeed, taggers are generally evaluated on perfectly tokenized texts where multiword units (MWU) have already been identified.

Our paper presents a MWU-aware POS tagger (i.e. a POS tagger including MWU recognition[1]). It is based on a Conditional Random Field (CRF) model that integrates features computed from large-coverage morphosyntactic lexicons and fine-grained MWU resources. We implemented it in a finite-state framework composed of a finite-state lexical analyzer and a CRF-decoder using weighted transducer composition.

In section 2, we will first describe statistical tagging based on CRF. Then, in section 3, we will show how to adapt the tagging models in order to also identify multiword unit. Next, section 4 will present the finite-state framework used to implement the tagger. Section 5 will focus on the description of our working corpus and the set of lexical resources used. In section 6, we then evaluate our tagger on French.

## 2 Statistical POS tagging with Linear Chain Conditional Random Fields

Linear chain Conditional Ramdom Fields (CRF) are discriminative probabilistic models introduced by (Lafferty et al., 2001) for sequential labelling. Given an input sequence $x = (x_1, x_2, ..., x_N)$ and an out-

[1] This strategy somewhat resembles the popular approach of joint word segmentation and part-of-speech tagging for Chinese, e.g. (Zhang and Clark, 2008). Moreover, other similar experiments on the same task for French are reported in (Constant et al., 2011).

put sequence of labels $y = (y_1, y_2, ..., y_N)$, the model is defined as follows:

$$P(y|x) = \frac{1}{Z(x)} \prod_{t=1}^{N} \exp\Big(\sum_{k=1}^{K} \lambda_k f_k(t, x, y_t, y_{t-1})\Big)$$

where $Z(x)$ is a normalization factor depending on $x$. It is based on $K$ features each of them being defined by a binary function $f_k$ depending on the current position $t$ in $x$, the current label $y_t$, the preceding one $y_{t-1}$ and the whole input sequence $x$. The feature is activated if a given configuration between $t$, $y_t$, $y_{t-1}$ and $x$ is satisfied (i.e. $f_k(t, y_t, y_{t-1}, x) = 1$). Each feature $f_k$ is associated with a weight $\lambda_k$. The weights are the parameters of the model. They are estimated during the training process by maximizing the conditional loglikelihood on a set of examples already labeled (training data). The decoding procedure consists in labelling a new input sequence with respect to the model, by maximizing $P(y|x)$ (or minimizing $-logP(y|x)$). There exist dynamic programming procedures such as Viterbi algorithm in order to efficiently explore all labelling possibilities.

Features are defined by combining different properties of the tokens in the input sequence and the labels at the current position and the preceding one. Properties of tokens can be either binary or textual: e.g. token contains a digit, token is capitalized (binary property), form of the token, suffix of size 2 of the token (textual property). Most taggers exclusively use language-independent properties – e.g. (Ratnaparkhi, 1996; Toutanova et al., 2003; Giménez and Márquez, 2004; Tsuruoka et al., 2009). It is also possible to integrate language-dependant properties computed from an external broad-coverage morphosyntactic lexicon, that are POS tags found in the lexicon for the given token (e.g. (Denis and Sagot, 2009)). It is of great interest to deal with unknown words[2] as most of them are covered by the lexicon, and to somewhat filter the list of candidate tags for each token. We therefore added to our system a language-dependent property: a token is associated with the concatenation of its possible tags in an external lexicon, i.e. the ambibuity class of the token ($AC$).

[2]Unknown words are words that did not occur in the training data.

In practice, we can divide features $f_k$ in two families: while *unigram features* ($u_k$) do not depend on the preceding tag, i.e. $f_k(t, y_t, y_{t-1}, x) = u_k(t, y_t, x)$, *bigram features* ($b_k$) depend on both current and preceding tags, i.e. $f_k(t, y_t, y_{t-1}, x) = b_k(t, y_t, y_{t-1}, x)$. In our practical case, bigrams exlusively depends on the two tags, i.e. they are independent from the input sequence and the current position like in the Hidden Markov Model (HMM)[3]. Unigram features can be sub-divided into internal and contextual ones. Internal features provide solely characteristics of the current token $w_0$: lexical form (i.e. its character sequence), lowercase form, suffice, prefix, ambiguity classes in the external lexicons, whether it contains a hyphen, a digit, whether it is capitalized, all capitalized, multiword. Contextual features indicate characteristics of the surroundings of the current token: token unigrams at relative positions -2,-1,+1 and +2 ($w_{-2}$, $w_{-1}$, $w_{+1}$,$w_{+2}$); token bigrams $w_{-1}w_0$, $w_0w_{+1}$ and $w_{-1}w_{+1}$; ambiguity classes at relative positions -2,-1,+1 and +2 ($AC_{-2}$, $AC_{-1}$, $AC_{+1}$,$AC_{+2}$). The different feature templates used in our tagger are given in table 2.

| | |
|---|---|
| **Internal unigram features** | |
| $w_0 = X$ | $\&t_0 = T$ |
| Lowercase form of $w_0 = L$ | $\&t_0 = T$ |
| Prefix of $w_0 = P$ with $\|P\| < 5$ | $\&t_0 = T$ |
| Suffix of $w_0 = S$ with $\|S\| < 5$ | $\&t_0 = T$ |
| $w_0$ contains a hyphen | $\&t_0 = T$ |
| $w_0$ contains a digit | $\&t_0 = T$ |
| $w_0$ is capitalized | $\&t_0 = T$ |
| $w_0$ is all capital | $\&t_0 = T$ |
| $w_0$ is capitalized and BOS[4] | $\&t_0 = T$ |
| $w_0$ is multiword | $\&t_0 = T$ |
| Lexicon tags $AC_0$ of $w_0 = A$ & $w_0$ is multiword | $\&t_0 = T$ |
| **Contextual unigram features** | |
| $w_i = X, i \in \{-2, -1, 1, 2\}$ | $\&t_0 = T$ |
| $w_iw_j = XY, (j, k) \in \{(-1, 0), (0, 1), (-1, 1)\}$ | $\&t_0 = T$ |
| $AC_i = A$ & $w_i$ is multiword, $i \in \{-2, -1, 1, 2\}$ | $\&t_0 = T$ |
| **Bigram features** | |
| $t_{-1} = T'$ | $\&t_0 = T$ |

Table 1: Feature templates

## 3 MWU-aware POS tagging

MWU-aware POS tagging consists in identifying and labelling lexical units including multiword ones.

[3]Hidden Markov Models of order $n$ use strong independance assumptions: a word only depends on its corresponding tag, and a tag only depends on its $n$ previous tags. In our case, $n$=1.

It is somewhat similar to segmentation tasks like chunking or Named Entity Recognition, that identify the limits of chunk or Named Entity segments and classify these segments. By using an IOB[5] scheme (Ramshaw and Marcus, 1995), this task is then equivalent to labelling simple tokens. Each token is labeled by a tag in the form X+B or X+I, where X is the POS labelling the lexical unit the token belongs to. Suffix B indicates that the token is at the beginning of the lexical unit. Suffix I indicates an internal position. Suffix O is useless as the end of a lexical unit corresponds to the beginning of another one (suffix B) or the end of a sentence. Such procedure therefore determines lexical unit limits, as well as their POS.

A simple approach is to relabel the training data in the IOB scheme and to train a new model with the same feature templates. With such method, most of multiword units present in the training corpus will be recognized as such in a new text. The main issue resides in the identification of unknown multiword units. It is well known that statistically inferring new multiword units from a rather small training corpus is very hard. Most studies in the field prefer finding methods to automatically extract, from very large corpus, multiword lexicons, e.g. (Dias, 2003; Caseli et al., 2010), to be integrated in Natural Language Processing tools.

In order to improve the number of new multiword units detected, it is necessary to plug the tagger to multiword resources (either manually built or automatically extracted). We incorporate new features computed from such resources. The resources that we use (cf. section 5) include three exploitable features. Each MWU encoded is obligatory assigned a part-of-speech, and optionally an internal surface structure and a semantic feature. For instance, the organization name *Banque de Chine* (Bank of China) is a proper noun (NPP) with the semantic feature ORG; the compound noun *pouvoir d'achat* (purchasing power) has a syntactic form NPN because it is composed of a noun (N), a preposition (P) and a noun (N). By applying these resources to texts, it is therefore possible to add four new properties for each token that belongs to a lexical multiword unit: the part-of-speech of the lexical multiword unit (POS), its internal structure (STRUCT), its semantic feature (SEM) and its relative position in the IOB scheme (POSITION). Table 2 shows the encoding of these properties in an example. The property extraction is performed by a longest-match context-free lookup in the resources. From these properties, we use 3 new unigram feature templates shown in table 3: (1) one combining the MWU part-of-speech with the relative position; (2) another one depending on the internal structure and the relative position and (3) a last one composed of the semantic feature.

| FORM | POS | STRUCT | POSITION | SEM | *Translation* |
|---|---|---|---|---|---|
| un | - | - | O | - | *a* |
| gain | - | - | O | - | *gain* |
| de | - | - | O | - | *of* |
| pouvoir | NC | NPN | B | - | *purchasing* |
| d' | NC | NPN | I | - | |
| achat | NC | NPN | I | - | *power* |
| de | - | - | O | - | *of* |
| celles | - | - | O | - | *the ones* |
| de | - | - | O | - | *of* |
| la | - | - | O | - | *the* |
| Banque | NPP | - | B | ORG | *Bank* |
| de | NPP | - | I | ORG | *of* |
| Chine | NPP | - | I | ORG | *China* |

Table 2: New token properties depending on Multiword resources

| **New internal unigram features** | |
|---|---|
| $POS_0/POSITION_0$ | $\&t_0 = T$ |
| $STRUCT_0/POSITION_0$ | $\&t_0 = T$ |
| $SEM_0$ | $\&t_0 = T$ |

Table 3: New features based on the MW resources

## 4 A Finite-state Framework

In this section, we describe how we implemented a unified Finite-State Framework for our MWU-aware POS tagger. It is organized in two separate classical stages: a preliminary resource-based lexical analyzer followed by a CRF-based decoder. The lexical analyzer outputs an acyclic finite-state transducer (noted TFST) representing candidate tagging sequences for a given input. The decoder is in charge of selecting the most probable one (i.e. the path in the TFST which has the best probability).

[5]I: Inside (segment); O: Outside (segment); B: Beginning (of segment)

### 4.1 Weighted finite-state transducers

Finite-state technology is a very powerful machinery for Natural Language Processing (Mohri, 1997; Kornai, 1999; Karttunen, 2001), and in particular for POS tagging, e.g. (Roche and Schabes, 1995). It is indeed very convenient because it has simple factorized representations and interesting well-defined mathematical operations. For instance, weighted finite-state transducers (WFST) are often used to represent probabilistic models such as Hidden Markov Models. In that case, they map input sequences into output sequences associated with weights following a probability semiring ($\mathbb{R}_+$,+,$\times$, 0, 1) or a log semiring ($\mathbb{R} \cup \{-\infty, +\infty\}$,$\oplus_{log}$,+, $+\infty$, 0) for numerical stability[6]. A WFST is a finite-state automaton which each transition is composed of an input symbol, an output symbol and a weight. A path in a WFST is therefore a sequence of consecutive transitions of the WFST going from an initial state to a final state, i.e. it puts a binary relation between an input sequence and an output sequence with a weight that is the product of the weights of the path transitions in a probability semiring (the sum in the log semiring). Note that a finite-state transducer is a WFST with no weights. A very nice operation on WFSTs is composition (Salomaa and Soittola, 1978). Let $T_1$ be a WFST mapping an input sequence $x$ into an output sequence $y$ with a weight $w_1(x, y)$, and $T_2$ be another WFST mapping a sequence $y$ into a sequence $z$ with a weight $w_2(y, z)$. The composition of $T_1$ with $T_2$ results in a WFST $T$ mapping $x$ into $z$ with a weight $w_1(x, y).w_2(y, z)$ in the probability semiring ($w_1(x, y) + w_2(y, z)$ in the log semiring).

### 4.2 Lexical analysis and decoding

The lexical analyzer is driven by lexical resources represented by finite-state transducers like in (Silberztein, 2000) (cf. section 5) and generates a TFST containing candidate analyses. Transitions of the TFST are labeled by a simple token (as input) and a POS tag (as output). This stage allows for reducing the global ambiguity of the input sentence in two different ways: (1) tag filtering, i.e. each token is only assigned its possible tags in the lexical resources; (2) segment filtering, i.e. we only keep lexical multiword units present in the resources. This implies the use of large-coverage and fine-grained lexical resources.

The decoding stage selects the most probable path in the TFST. This involves that the TFST should be weighted by CRF-based probabilities in order to apply a shortest path algorithm. Our weighing procedure consists in composing a WFST encoding the sentence unigram probabilities (unigram WFST) and a WFST encoding the bigram probabilities (bigram WFST). The two WFSTs are defined over the log semiring. The unigram WFST is computed from the TFST. Each transition corresponds to a ($x_t$,$y_t$) pair at a given position $t$ in the sentence $x$. So each transition is weighted by summing the weights of the unigram features activated at this position. In our practical case, bigram features are independent from the sentence $x$. The bigram WFST can therefore be constructed once and for all for the whole tagging process, in the same way as for order-1 HMM *transition* diagrams (Nasr and Volanschi, 2005).

## 5 Linguistic resources

### 5.1 French TreeBank

The French Treebank (FTB) is a syntactically annotated corpus[7] of 569,039 tokens (Abeillé et al., 2003). Each token can be either a punctuation marker, a number, a simple word or a multiword unit. At the POS level, it uses a tagset of 14 categories and 34 sub-categories. This tagset has been optimized to 29 tags for syntactic parsing (Crabbé and Candito, 2008) and reused as a standard in a POS tagging task (Denis and Sagot, 2009). Below is a sample of the FTB version annotated in POS.

| | | |
|---|---|---|
| , | PONCT | , |
| soit | CC | *i.e.* |
| une | DET | *a* |
| augmentation | NC | *raise* |
| de | P | *of* |
| 1_ _2 | DET | *1_ _2* |
| % | NC | *%* |
| par_rapport_au | P+D | *compared with the* |
| mois | NC | *preceding* |
| précédent | ADJ | *month* |

[6] A semiring $\mathbb{K}$ is a 5-tuple $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$ where the set $\mathbb{K}$ is equipped with two operations $\oplus$ and $\otimes$; $\bar{0}$ and $\bar{1}$ are their respective neutral elements. The log semiring is an image of the probability semiring via the $-log$ function.

[7] It is made of journalistic texts from *Le Monde* newspaper.

Multiword tokens encode multiword units of different types: compound words and named entities. Compound words mainly include nominals such as *acquis sociaux* (social benefits), verbs such as *faire face à* (to face) adverbials like *dans l' immédiat* (right now), prepositions such as *en dehors de* (beside). Some Named Entities are also encoded: organization names like *Société suisse de microélectronique et d' horlogerie*, family names like *Strauss-Kahn*, location names like *Afrique du Sud* (South Africa) or *New York*. For the purpose of our study, this corpus was divided in three parts: 80% for training (TRAIN), 10% for development (DEV) and 10% for testing (TEST).

### 5.2 Lexical resources

The lexical resources are composed of both morphosyntactic dictionaries and strongly lexicalized local grammars. Firstly, there are two general-language dictionaries of simple and multiword forms: DELA (Courtois, 1990; Courtois et al., 1997) and Lefff (Sagot, 2010). DELA has been developped by a team of linguists. Lefff has been automatically acquired and then manually validated. It also resulted from the merge of different lexical sources. In addition, we applied specific manually built lexicons: Prolex (Piton at al., 1999) containing toponyms ; others including organization names and first names (Martineau et al., 2009). Figures on these dictionaries are detailed in table 4.

| Name | # simple forms | #MW forms |
|---|---|---|
| DELA | 690,619 | 272,226 |
| Lefff | 553,140 | 26,311 |
| Prolex | 25,190 | 97,925 |
| Organizations | 772 | 587 |
| First names | 22,074 | 2,220 |

Table 4: Morphosynctatic dictionaries

This set of dictionaries is completed by a library of strongly lexicalized local grammars (Gross, 1997; Silberztein, 2000) that recognize different types of multiword units such as Named Entities (organization names, person names, location names, dates), locative prepositions, numerical determiners. A local grammar is a graph representing a recursive finite-state transducer, which recognizes sequences belonging to an algebraic language. Practically, they describe regular grammars and, as a consequence, can be compiled into equivalent finite-state transducers. We used a library of 211 graphs. We manually constructed from those available in the online library GraalWeb (Constant and Watrin, 2007).

### 5.3 Lexical resources vs. French Treebank

In this section, we compare the content of the resources described above with the encodings in the FTB-DEV corpus. We observed that around 97,4% of lexical units encoded in the corpus (excluding numbers and punctuation markers) are present in our lexical resources (in particular, 97% are in the dictionaries). While 5% of the tokens are unknown (i.e. not present in the training corpus), 1.5% of tokens are unknown and not present in the lexical resources, which shows that 70% of unknown words are covered by our lexical resources.

The segmentation task is mainly driven by the multiword resources. Therefore, they should match as much as possible with the multiword units encoded in the FTB. Nevertheless, this is practically very hard to achieve because the definition of MWU can never be the same between different people as there exist a continuum between compositional and non-compositional sequences. In our case, we observed that 75.5% of the multiword units in the FTB-DEV corpus are in the lexical resources (87.5% including training lexicon). This means that 12.5% of the multiword tokens are totally unknown and, as a consequence, will be hardly recognized. Another significant issue is that many multiword units present in our resources are not encoded in the FTB. For instance, many Named Entities like dates, person names, mail addresses, complex numbers are absent. By applying our lexical resources[8] in a longest-match context-free manner with the platform Unitex (Paumier, 2011), we manually observed that 30% of the multiword units found were not considered as such in the FTB-DEV corpus.

## 6 Experiments and Evaluation

We firstly evaluated our system for standard tagging without MWU segmentation and compare it with other available statistical taggers that we all trained on the FTB-TRAIN corpus. We tested the

[8]We excluded local grammars recognizing dates, person names and complex numbers.

well-known TreeTagger (Schmid, 1994) based on probabilistic decision trees, as well as TnT (Brants, 2000) implementing second-order Hidden Markov. We also compared our system with two existing discriminative taggers: SVMTool (Giménez and Márquez, 2004) based on Support Vector Models with language-independent features; MElt (Denis and Sagot, 2009) based on a Maximum Entropy model also incorporating language-dependent feature computed from an external lexicon. The lexicon used to train and test MElt included all lexical resources[9] described in section 5. For our CRF-based system, we trained two models with CRF++[10]: (a) STD using language-independent template features (i.e. excluding $AC$-based features); (b) LEX using all feature templates described in table 2. We note CRF-STD and CRF-LEX the two related taggers when no preliminary lexical analysis is performed; CRF-STD+ and CRF-LEX+ when a lexical analysis is performed. The lexical analysis in our experiment consists in assigning for each token its possible tags found in the lexical resources[11]. Tokens not found in the resources are assigned all possible tags in the tagset in order to ensure the system robustness. If no lexical analysis is applied, our system constructs a TFST representing all possible analyzes over the tagset. The results obtained on the TEST corpus are summed up in table 5. Column ACC indicates the tagger accuracy in percentage. We can observe that our system (CRF-LEX+) outperforms the other existing taggers, especially MElt whose authors claimed state-of-the-art results for French. We can notice the great interest of a lexical analysis as CRF-STD+ reaches similar results as a MaxEnt model based on features from an external lexicon.

We then evaluated our MWU-aware tagger trained on the TRAIN corpus whose complex tokens have been decomposed in a sequence of simple tokens and relabeled in the IOB representation. We used three different sets of feature templates leading to three CRF models: CRF-STD, CRF-LEX and CRF-MWE. The two first ones (STD and LEX) use the same feature templates as in the previous experiment. MWE includes all feature templates decribed in sections 2 and 3. CRF-MWE+ indicates that a preliminary lexical analysis is performed before applying CRF-MWE decoding. The lexical analysis is achieved by assigning all possible tags of simple tokens found in our lexical resources, as well as adding, in the TFST, new transitions corresponding to MWU segments found in the lexical resources. We compared the three models with a baseline and SVMTool that have been learnt on the same training corpus. The baseline is a simple context-free lookup in the training MW lexicon, after a standard CRF-based tagging with no MW segmentation. We evaluated each MWU-aware tagger on the decomposed TEST corpus and computed the f-score, combining precision and recall[12]. The results are synthesized in table 6. The SEG column shows the segmentation $f$-score solely taking into account the segment limits of the identified lexical unit. The TAG column also accounts for the label assigned. The first observation is that there is a general drop in the performances for all taggers, which is not a surprise as regards with the complexity of MWU recognition (97.7% for the best standard tagger vs. 94.4% for the best MWU-aware tagger). Clearly, MWU-aware taggers which models incorporate features based on external MWU resources outperform the others. Nevertheless, the scores for the identification and the tagging of the MWUs are still rather low: 91%-precision and 71% recall. We can also see that a preliminary lexical analysis slightly lower the scores, which is due to

| Tagger | Model | ACC |
|---|---|---|
| TnT | HMM | 96.3 |
| TreeTagger | Decision trees | 96.4 |
| SVMTool | SVM | 97.2 |
| CRF-STD | CRF | 97.4 |
| MElt | MaxEnt | 97.6 |
| CRF-STD+ | CRF | 97.6 |
| **CRF-LEX** | CRF | **97.7** |
| **CRF-LEX+** | CRF | **97.7** |

Table 5: Comparison of different taggers for French

[9] Dictionaries were all put together, as well as with the result of the application of the local grammars on the corpus.

[10] CRF++ is an open-source toolkit to train and test CRF models (http://crfpp.sourceforge.net/). For training, we set the cut-off threshold for features to 2 and the C value to 1. We also used the L2 regularization algorithm.

[11] Practically, as the tagsets of the lexical resources and the FTB were different, we had to first map tags used in the dictionaries into tags belonging to the FTB tagset.

[12] f-score $f = \frac{2pr}{p+r}$ where $p$ is precision and $r$ is recall.

missing MWUs in the resources and is a side effect of missing encodings in the corpus.

| Tagger | Model | TAG | SEG |
|---|---|---|---|
| Baseline | CRF | 91.2 | 93.6 |
| SVMTool | SVM | 92.1 | 94.7 |
| CRF-STD | CRF | 93.7 | 95.8 |
| CRF-LEX | CRF | 93.9 | 95.9 |
| **CRF-MWE** | CRF | **94.4** | **96.4** |
| CRF-MWE+ | CRF | 94.3 | 96.3 |

Table 6: Evaluation of MWU-aware tagging

With respect to the statistics given in section 5.3, it appears clearly that the evaluation of MWU-aware taggers is somewhat biased by the fact that the definition of the multiword units encoded in the FTB and the ones listed in our lexical resources are not exactly the same. Nevertheless, this evaluation that is the first in this context, brings new evidences on the importance of multiword unit resources for MWU-aware tagging.

## 7 Conclusions and Future Work

This paper presented a new part-of-speech tagger including multiword unit identification. It is based on a CRF model integrating language-independent features, as well as features computed from external lexical resources. It was implemented in a finite-state framework composed of a preliminary finite-state lexical analysis and a CRF decoding using weighted finite-state transducer composition. The tagger is freely available under the LGPL license[13]. It allows users to incorporate their own lexicons in order to easily integrate it in their own applications.

We showed that the tagger reaches state-of-the-art results for French in the standard evaluation environment (i.e. each multiword unit is already merged in a single token). The evaluation of the tagger integrating MWU recognition clearly shows the interest of incorporating features based on MWU resources. Nevertheless, as there exist some differences in the MWU definitions between the lexical resources and the working corpus, this first experiment requires further investigations. First of all, we could test our tagger by incorporating lexicons of MWU automatically extracted from large raw corpora in order to deal with low recall. We could as well combine the lexical analyzer with a Named Entity Recognizer. Another step would be to modify the annotations of the working corpus in order to cover all MWU types and to make it more homogeneous with our definition of MWU. Another future work would be to test semi-CRF models that are well-suited for segmentation tasks.

[13] http://igm.univ-mlv.fr/~mconstan/research/software

## References


A. Abeillé, L. Clément and F. Toussenel. 2003. Building a treebank for French. in A. Abeillé (ed), *Treebanks*, Kluwer, Dordrecht.

T. Brants. 2000. TnT - A Statistical Part-of-Speech Tagger. *In Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP 2000)*, 224–231.

H. Caseli, C. Ramisch, M. das Graas Volpe Nunes, A. Villavicencio. 2010. Alignment-based extraction of multiword expressions. *Language Resources and Evaluation*, Springer, vol. 44(1), 59–77.

M. Constant, I. Tellier, D. Duchier, Y. Dupont, A. Sigogne, S. Billot. 2011. Intégrer des connaissances linguistiques dans un CRF : application à l'apprentissage d'un segmenteur-étiqueteur du français. *In Actes de la Conférence sur le traitement automatique des langues naturelles (TALN'11)*.

M. Constant and P. Watrin. 2007. Networking Multiword Units. *In Proceedings of the 6th International Conference on Natural Language Processing (GoTAL'08)*, Lecture Notes in Artificial Intelligence, Springer-Verlag, vol. 5221: 120 – 125.

A. Copestake, F. Lambeau, A. Villavicencio, F. Bond, T. Baldwin, I. A. Sag and D. Flickinger. 2002. Multiword expressions: linguistic precision and reusability. *In Proceedings of the Third conference on Language Resources and Evaluation (LREC' 02)*, 1941 – 1947.

B. Courtois. 1990. *Un système de dictionnaires électroniques pour les mots simples du français*. Langue Française, vol. 87: 1941 – 1947.

B. Courtois, M. Garrigues, G. Gross, M. Gross, R. Jung, M. Mathieu-Colas, A. Monceaux, A. Poncet-Montange, M. Silberztein, R. Vivés. 1990. *Dictionnaire électronique DELAC : les mots composés binaires*. Technical report, LADL, University Paris 7, vol. 56.

B. Crabbé and M. -H. Candito. 2008. Expériences d'analyse syntaxique statistique du franais. *In Proceedings of Traitement des Langues Naturelles (TALN 2008)*.

P. Denis et B. Sagot. 2009. Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art

POS tagging with less human effort. *In Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation (PACLIC 2009)*.

G. Dias. 2003. Multiword Unit Hybrid Extraction. *In proceedings of the Workshop on Multiword Expressions of the 41st Annual Meeting of the Association of Computational Linguistics (ACL 2003)*, 41–49.

J. Giménez and L. Márquez. 2004. SVMTool: A general POS tagger generator based on Support Vector Machines. *In Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*.

M. Gross. 1997. The construction of local grammars. In E. Roche and Y. Schabes (eds.). *Finite-State Language Processing*. The MIT Press, Cambridge, Mass. 329–352

L. Karttunen. 2001. Applications of Finite-State Transducers in Natural Language Processing. *In proceedings of the 5th International Conference on Implementation and Application of Automata (CIAA 2000)*. Lecture Notes in Computer Science. vol. 2088, Springer, 34–46

A. Kornai (Ed.). 1999. *Extended Finite State Models of Language*. Cambridge University Press

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random Fields: Probabilistic models for segmenting and labeling sequence data. *In Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, 282–289.

C. Martineau, T. Nakamura, L. Varga and Stavroula Voyatzi. 2009. Annotation et normalisation des entités nommées. *Arena Romanistica*. vol. 4:234–243.

M. Mohri 1997. *Finite-state transducers in language and speech processing*. Computational Linguistics 23 (2):269–311.

A. Nasr, A. Volanschi. 2005. Integrating a POS Tagger and a Chunker Implemented as Weighted Finite State Machines. *Finite-State Methods and Natural Language Processing*, Lecture Notes in Computer Science, vol. 4002, Springer 167–178.

S. Paumier. 2011. *Unitex 2.1 user manual*. http://igm.univ-mlv.fr/˜unitex.

O. Piton, D. Maurel, C. Belleil. 1999. The Prolex Data Base : Toponyms and gentiles for NLP. *In proceedings of the Third International Workshop on Applications of Natural Language to Data Bases (NLDB'99)*, 233–237.

L. A. Ramshaw and M. P. Marcus. 1995. Text chunking using transformation-based learning. *In Proceedings of the 3rd Workshop on Very Large Corpora*, 88 – 94.

A. Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 1996)*, 133 – 142.

E. Roche, Y. Schabes. 1995. Deterministic part-of-speech tagging with finite-state transducers. *Computational Linguistics*, MIT Press, vol. 21(2), 227–253

I. A. Sag, T. Baldwin, F. Bond, A. Copestake, D. Flickinger. 2001. Multiword Expressions: A Pain in the Neck for NLP. *In Proceedings of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2002)*, 1–15

B. Sagot. 2010. The Lefff, a freely available, accurate and large-coverage lexicon for French. *In Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'10)*.

A. Salomaa, M. Soittola. 1978. *Automata-Theoretic Aspects of Formal Power Series*. Springer-Verlag.

H. Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. *Proceedings of International Conference on New Methods in Language Processing*.

M. Silberztein. 2000. INTEX: an FST toolbox. *Theoretical Computer Science*, vol. 231 (1): 33–46.

K. Toutanova, D. Klein, C. D. Manning, Y. Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. *Proceedings of HLT-NAACL 2003*, 252 – 259.

Y. Tsuruoka, J. Tsujii, S. Ananiadou. 2009. Fast Full Parsing by Linear-Chain Conditional Random Fields. *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009)*, 790–798.

Y. Zhang, S. Clark. 2008. Joint Word Segmentation and POS Tagging Using a Single Perceptron. *Proceedings of ACL 2008*, 888 – 896.